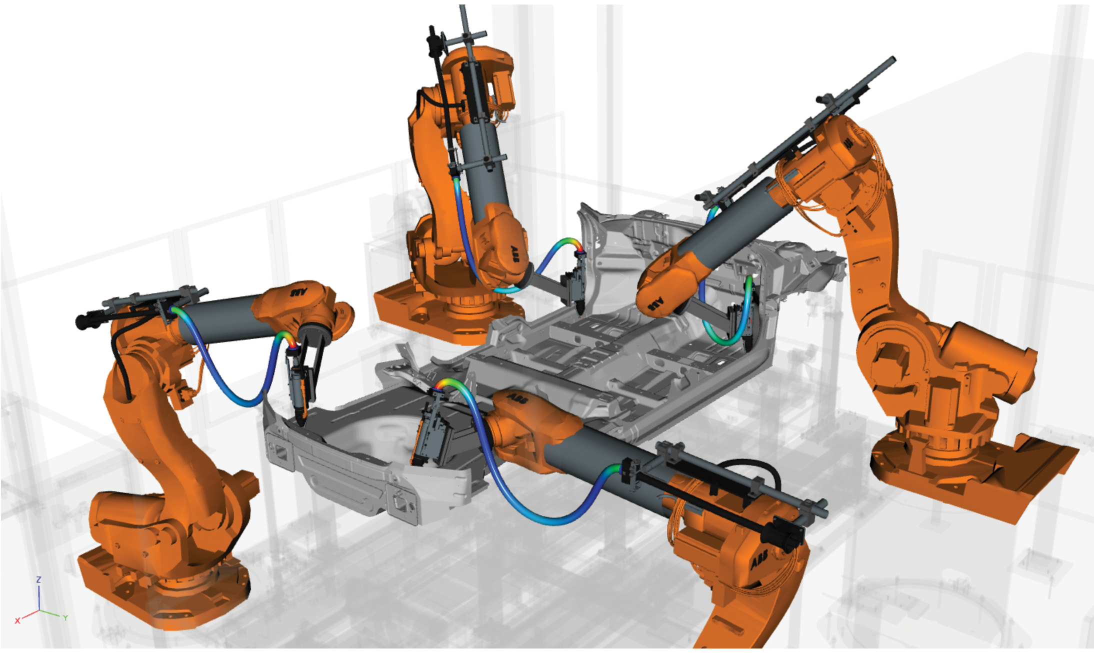
CHALMERS



Path Optimization for Multi-Robot Station Minimizing Dresspack Wear

Master of Science Thesis in Systems, Control and Mechatronics

JONAS KRESSIN

Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
Göteborg, Sweden 2013

Master's Thesis EX013/2013

Path Optimization for Multi-Robot Station Minimizing Dresspack Wear
JONAS KRESSIN

Master's Thesis EX013/2013
Department of Signals and Systems
Division of Automatic Control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31-772 1000

This Master's Thesis was carried out at Fraunhofer Chalmers Research Centre for Industrial Mathematics in Göteborg, Sweden.

Cover:
IPS simulation of stud welding station with dresspack bending moment analysis.

Abstract

When dealing with off-line programming of industrial robots there are sophisticated softwares available for planning of the robot paths, one being *Industrial Path Solutions (IPS)* developed by the Fraunhofer Chalmers Centre. One thing that is not taken into account when finding the robot paths is potential wearing on the robots' cable dresspacks. Since dresspacks wearing out is very expensive both in material cost and cost from downtime, there is a need for incorporating dresspack wear consideration when making the automatic path planning.

This thesis addresses the problem of finding robot paths that are less damaging for the dresspack, and the result consists of three different methods for dealing with this problem. The first method involves computationally efficient restrictions of the robot joint values in order to avoid damage to the dresspack. The second method deals with the issue of finding cable configurations that are robust to movements, since only robust configurations should be used in the final sequences. Finally, the third method involves a function that measures the cable wear as a cost, to then be minimized when doing the automatic path planning.

The three methods are tested and evaluated individually on a test case in IPS. The tests show that with the cable wear consideration, the robot takes different paths with lower values of the wearing measures than the case without cables. It is concluded that with some improvements of the methods, they can be combined into a fully implementable solution.

Keywords: cable wear minimization, robot cable simulation, path planning, joint restrictions, robust cable configurations, cable wear cost function.

Acknowledgements

First of all I would like to thank my supervisors Daniel Segerdahl, MSc, and Tomas Hermansson, MSc, for their most valuable input and support throughout this thesis work. I would also like to thank Robert Bohlin, Phd, for rewarding discussions on path planning and optimization, and Johan Carlson, Phd and Director, for his guidance and support throughout the work. I would also like to express my gratitude to my supervisor and examiner from Chalmers, professor Bengt Lennartson, for his interest in this thesis.

Further on, I want to thank Mathias Sundbäck, Mikael Andersson and Johan Segeborn at the Manufacturing Engineering department at Volvo Cars in Göteborg. Your welcoming attitude helped me gain invaluable insight of the challenges with dresspack wear, as well as rewarding on-site experience of the problems.

Last but not least I want to express my gratitude towards everyone else at FCC that has helped me. The including atmosphere and willingness to help has really contributed a lot to my work.

Göteborg May 2013
Jonas Kressin

Contents

Chapter 1

Introduction

This chapter presents the **background** to this Master's Thesis, including motivations from industry. It also states the **purpose and goal** as well as **delimitations**.

1.1 Background

In time demanding robotic applications it is of great interest to find an optimized way to perform a given task. One example is robot welding in car industry, where a given set of welds are to be done and a set of stations and robots are given to do the welds. The task is to find which robot should do which welds, and in what order, with a goal to minimize the total time. Fraunhofer-Chalmers Research Centre for Industrial Mathematics (FCC) has developed a path planning software called *Industrial Path Solutions (IPS)*, which among many other features performs exactly this task. In excess of this, IPS also supports simulation of flexible components (cables).

One thing that IPS has not been taking into account when doing the automatic path planning is potential wearing of the robot's cable dresspack. This wearing can be e.g. that

- the cable hits some static geometry (e.g. sharp sheet metal on the car).
- the cable is bent in a bad way.
- the cable gets stuck somewhere and then tugged.

Damaged cable dresspacks are very expensive, both due to high costs for buying new dresspacks and in particular stop in production. According to a study at Volvo Cars, 47% of the robot dresspacks wore out faster than the promised life length of one year [1]. Out of all dresspack related breakdowns, 61% were considered to be major, i.e. $\geq$ 30 min [2]. The study also showed an existing potential to improve the situation, with an estimation of 14% wear out instead of 47% if appropriate actions were to be taken. Besides this, the robotic cable protection company REIKU claims that *"Almost*

85% *of Robotics and Automation "downtime" can be directly attributed to cable or hose failure"* [3]. Also [4] and [5] report that failing cables is the foremost cause of downtime for industrial robots.

The study at Volvo Cars showed that for some robots the dresspack never wore out during the study period, whereas for others it wore out up to six times. Based on this and insights from matter experts, it was established that the root cause likely was that proper optimization of the robot path had never been performed [1]. Therefore, if the dresspack wear would be considered at an early stage of planning, that could have a significant effect on the robot breakdowns.

Modeling of robot dresspacks has been done in previous works, like e.g. in [6]. Here a cable was modeled on a roller hemming robot and various simulations were performed. The simulations included analyses of length, curvature, bending, tension and shearing. Although the simulations did include analyses related to wearing, the focus was on mounting and dresspack design rather than wearing minimization through path optimization.

1.2 Purpose and goal

The purpose of this Master's Thesis is to derive methods to minimize cable wear when doing the automatic off-line programming in IPS. Questions to be answered are:

- Do the methods perform as intended?
- In excess of these methods, what more is needed for a commercially acceptable solution?

The goal is to evaluate and verify functionality individually for each method, to then conclude whether the methods can be combined to minimize dresspack wear on a multi-robot station.

1.3 Delimitations

Since this Master's Thesis is part of a collaboration between other projects, and since some simulation features are currently not fully developed, the following is not included in the goal and proceedings:

- The mathematical modeling of a cable (already implemented in IPS).
- Compilation of the path planning algorithm (already implemented in IPS).
- Simulation of dynamical behavior of a cable (effects due to acceleration, not fully developed).

1.4 Summary

It has now been established that dresspack wear is a profound problem in industry, and that dresspack wear consideration in the automatic off-line programming could have a significant effect on robot breakdowns. To deal with the problem of cable wear, this thesis aims at deriving methods for minimizing cable wear when doing the automatic off-line programming, with the delimitations as stated in the previous section. Before the derivation of these methods, **Chapter 2, Theory** will provide a brief theory foundation with some general knowledge about robots, path planning and dresspacks.

Chapter 2

Theory

This chapter provides a brief theory foundation for this Master's Thesis. The purpose is to provide general knowledge on the topics of **robot dresspacks**, **cable modeling**, **robot kinematics**, **path planning** and **optimization**.

2.1 Robot dresspacks

For an industrial robot to be able to perform a task it needs some kind of *tool*. A tool can be e.g. a welding gun, a spray painting tool or a gripper. Each type of tool requires one or several types of resources, like e.g. electricity, pneumatics, material feeding or information exchange. To supply the tool with its resources there is a bundle of cables and hoses connecting to the tool, called a dresspack.

The conventional way of routing a dresspack is externally along the upper arm, *external dressing* [7]. This routing is best suited for installations with low performance and low wrist movement complexity [8], but is still common among higher performance applications. The problem with the external dressing is that it occupies space along the robot arm, which increases the possibility of collision with surrounding geometry. Another problem is the swinging motions of the dresspack, which causes wearing [7]. As a complement to the external dressing, *internal dressing* or *integrated dresspack* has emerged on the market. The internal dressing runs inside the robot upper arm and through the robot wrist, occupying much less space than the external. This makes the offline programming much easier, since the robot movements no longer need to be restricted because of the dresspack [8]. Also, since the swinging movements are avoided the wearing is significantly decreased [8].

2.2 Cable modeling

To simulate the dresspacks as slender, flexible objects, a mahematical model of a cable is needed. A cable or hose can be modeled as a slender one dimensional elastic object with undeformed cross section, for both large and small deformations [9]. The characteristic deformed shape of a cable is captured in a so called *Cosserat rod*, which can be seen as a slender beam. The Cosserat rod is parameterized by arc length s (see Figure 2.1), and is defined by the frames $R(s) = (d_1,d_2,d_3)$ defining the cross section orientations, and a center curve $\varphi(s)$ going through the center of the cross sections. The frame vectors d_1, d_2 and d_3 are orthonormal; d_1 and d_2 span the cross-section plane and d_3 is the cross-section normal. To acquire the deformed shape, each material point in the un-deformed cable is mapped to the deformed via the deformation mapping

$$\chi : [0,L] \times \mathcal{A} \mapsto \mathbb{R}^3 \tag{2.1}$$

where

$$\chi(s,\xi_1,\xi_2) = \varphi(s) + \xi_1 \cdot d_1(s) + \xi_2 \cdot d_2(s) \tag{2.2}$$

Here $\mathcal{A}$ is the cross-section and ξ_1,ξ_2 are planar coordinates in $\mathcal{A}$.

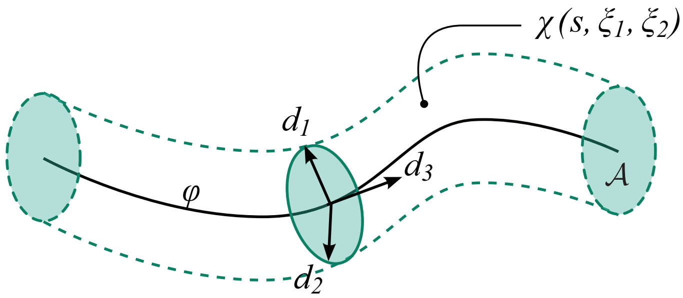


Figure 2.1: A cable segment represented as a Cosserat rod [10].

2.3 Robot kinematics

Robot kinematics is about controlling the positions, velocities and accelerations of the links of a manipulator, which in the case of an industrial robot is a robot arm [11]. The links are the rigid bodies that the arm is built up by, and each link is manipulated by a revolute joint. Link zero (the base) is static, link one is attached to joint one, link two to joint two etc. Figure 2.2 shows a typical joint setup for a six axis industrial robot, where link n connects joints n and $n + 1$ for $n = 1...5$. The last link connects joint six to a tool attachment location called a *tool plate*.

To acquire the position and orientation of the tool plate relative to the robot base, a method of multiplying transformation matrices is used [11]. The relation between link

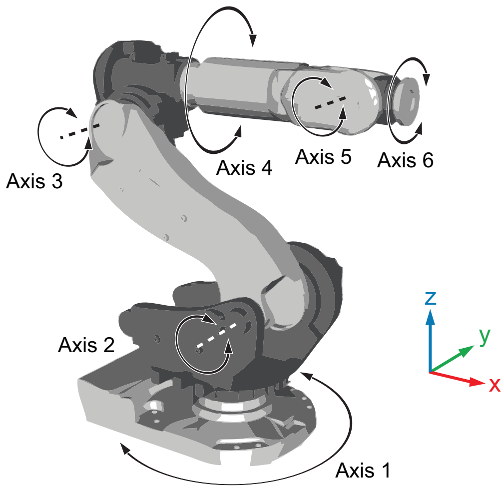


Figure 2.2: Typical joint setup for an industrial robot with six axes (joints).

n and $n+1$ is described by the transformation matrix

$$^{n}T_{n+1} = \begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

which consists of one rotation (x,y,z) and one translation (p_x,p_y,p_z) part. The transformation matrix of the tool plate, called the *hand frame (H)*, relative to the base *(R)* is then obtained as

$$^{R}T_H = {}^{R}T_1\,{}^{1}T_2 \ldots {}^{n-2}T_{n-1}\,{}^{n-1}T_H \tag{2.4}$$

As an example, consider again the robot in Figure 2.2. The transformation matrix for link two is obtained as

$$\begin{aligned} {}^{R}T_2 = {}^{R}T_1{}^{1}T_2 &= [TRANS(x,l_1)ROT(z,j_1)][TRANS(z,l_2)TRANS(x,l_3)ROT(y,j_2)] \\ &= \begin{bmatrix} \cos(j_1) & -\sin(j_1) & 0 & l_1 \\ \sin(j_1) & \cos(j_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(j_2) & 0 & \sin(j_2) & l_3 \\ 0 & 1 & 0 & 0 \\ -\sin(j_2) & 0 & \cos(j_2) & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(j_1)\cos(j_2) & -\sin(j_1) & \cos(j_1)\sin(j_2) & l_3\cos(j_1)+l_1 \\ \sin(j_1)\cos(j_2) & \cos(j_1) & \sin(j_1)\sin(j_2) & l_3\sin(j_1) \\ -\sin(j_2) & 0 & \cos(j_2) & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.5}$$

where l_1, l_2, l_3 are lengths and j_1, j_2 are joint angles. The method described above is referred to as *direct kinematics*, i.e. finding the transformation matrix of the hand frame given the robot joint values. To do the opposite, i.e. to find the joint values given the hand frame transformation, is called solving the *inverse kinematics* and is more complicated. There are several different algorithms available to solve the inverse kinematics, one heuristic being to put the manipulator transformation matrix ${}^{R}T_{H}$ equal to the general transformation matrix in (2.5), and then solving for specific elements. However there might not be one single solution for a specific transformation, but rather several solutions. This is called *redundancy*, and is a common occurrence when solving the inverse kinematics. There can also be infinitely many solutions; this is referred to as *degeneracy.*

2.4 Path planning

With a description of how the kinematics affect the robot position, a framework for planning of the robot paths can now be derived. To understand what path planning is one can consider the *Piano Mover's Problem* [12]. Consider a 2-or 3-dimensional drawing or model of a house and a piano. The problem is to move the piano from one room to another in an efficient way, without colliding with anything. In robotics this problem translates to moving the robot from one configuration to another without colliding with any static geometry or with the robot itself.

To reduce complexity and pave the way for further planning theory, the robot configurations are represented in a *configuration space (C-space)* [12]. A configuration expresses the position of the robot in terms of its joint angles, and given a sample point in C-space it is possible to check whether the robot is in collision or not. With the C-space representation in order, the problem is now reduced to finding a collision free path from a start configuration, q_{init}, to a goal configuration, q_{goal}, or determine that no such path exists.

One method for finding a collision free path between q_{init} and q_{goal}, is the *Probabilistic Roadmap Method (PRM)* [13]. The main idea is to first do a preprocessing step to acquire a network *(roadmap)* of randomly distributed collision free configurations, and then to connect q_{init} and q_{goal} via the network nodes (see Figure 2.3). If there is no possible way of connecting the configurations without colliding with any obstacles, the method has failed to find a feasible path and a denser sampling is needed. It is shown in [13] that as time tends to infinity and the number of sample points increases, the probability of classifying a feasible problem as infeasible tends to zero. This is called *probabilistic completeness* and is a key feature for a planning algorithm.

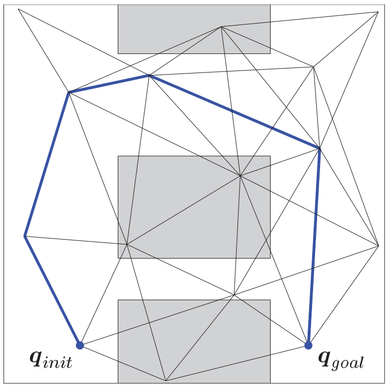


Figure 2.3: Example of a probabilistic roadmap from q_{init} to q_{goal} [13].

2.5 Optimization

As described in Section 2.4, the path planning procedure involves random distribution of points in the configuration space. Since this randomness might lead to strange, "jerky" paths for the robot, there is a need for some kind of smoothing of the paths. In [13] a smoothing procedure is suggested, where new points are added around the acquired path, upon which a shorter feasible path is searched for locally around the old one. When also considering e.g. time, and eventually cable wear, this smoothing procedure can be seen as a multi-objective optimization problem.

The term "to optimize" is explained in [14] as *"to do something as well as is possible"*. In mathematical terms this translates into altering a set of *variables*, in order to *minimize* or *maximize* an *objective function*, without violating certain *constraints*. When dealing with minimization, the objective function is often referred to as a *cost function*. With this terminology the problem can intuitively be translated into *to do something with as low cost as possible* or *to do something as cheap as possible*. For the smoothing problem, the cost consists of several sub-costs, like e.g. traveling distance cost and time cost. The objective is then to minimize the total cost by selecting sample points in the configuration space.

When selecting the sample points, it makes sense to select points that result in a lower value for the cost function. The direction in the configuration space that gives a lower cost is called a *direction of descent*. A sufficient condition for descent, given in [14], is that for a cost function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ in C^1 around a point $\mathbf{x}$, for which $f(\mathbf{x}) < +\infty$, there exists a vector $\mathbf{p} \in \mathbb{R}^n$ such that

$$\nabla f(\mathbf{x})^T \mathbf{p} < 0 \tag{2.6}$$

holds. $\mathbf{p}$ is then a descent direction with respect to f at $\mathbf{x}$. An intuitive and graphical interpretation of this is to move in the opposite direction of the gradient of the cost function, or rather in a direction $> 90°$ away from the gradient.

2.6 Summary

The topic of path optimization for robots considering dresspack wear covers several areas of expertise, of which a significant part has been briefly presented in this chapter. The most important parts for further understanding of this thesis are the robot kinematics and path planning and optimization; in particular the robot joint setup and the understanding of having C-space configurations as variables for the optimization. With a theory foundation in order, the problem of finding robot paths with minimized dresspack wear can now be approached in **Chapter 3, Methodology**.

Chapter 3

Methodology

This chapter presents the methods and procedures used in this Master's Thesis. The structure follows a chronological order, starting with **finding causes for cable wear**, then finding **methods for implementing solutions** and finally formulating a methodology for a **test case evaluation**.

3.1 Finding causes for cable wear

In order to detect and avoid cable wear, it was essential to acquire knowledge about the cause of the wearing. To get a broad picture, interviews and discussions were held with simulation experts and robot programmers with hands-on experience from cable wear. Since the framework for this Master's Thesis was to perform path optimization, the causes needed to be filtered out to only include the ones that could be affected by the robot path. This excluded wearing causes like e.g. poor rigging of the dresspacks and badly chosen dresspack types for a given robot application. These kinds of wearing causes cannot be dealt with through path optimization, but need to be addressed by the dresspack designers.

Further on, the studies in [1] and [2] pointed out and confirmed the causes acquired from the interviews. In connection with the interviews, company visits to Volvo Cars were made, at which a deeper knowledge and understanding of the problems was acquired.

3.2 Methods for implementing solutions

To achieve the main goal of finding paths with reduced cable wear, the work was divided into different subtasks. The interviews revealed that some problems were best solved with path optimization considering cable wear, but also that some wearing could be avoided with computationally efficient restrictions on the robot joint values. It was also established that some method for determining robustness of configurations was needed.

3.2.1 Joint restrictions

From the interviews it was found that some hands-on solutions, or ”rules of thumb”, that the robot simulators at Volvo were using could be translated into computationally efficient rules for joint restrictions. This meant that some wearing could be avoided by restricting the values of the robot joint angles. To translate the ”rules of thumb” into computer implementable rules, an understanding of how the joint values affect the cable needed to be found. This understanding lead to geometrical relationships between the joint values and the cable shape, which in turn could be modeled using mathematical geometry.

3.2.2 Robust cable configurations

When performing automatic path planning without any cables, the location of the robot is *deterministic*, given a certain robot configuration. This is because the robot consists of rigid bodies and the surrounding geometry is static. However when attaching a cable to a robot, the shape of the cable is *not* deterministic, given a robot configuration. The shape might depend on the previous traveling path of the cable.

In the first step of the automatic path planning, IPS finds the set of all possible robot configurations for reaching a weld point. Since the path to reaching such a point at this stage has not been decided yet, it was necessary to only include points where the cable shape could be determined regardless of previous traveling path. A cable configuration is specified by the rotation and translation of the cable’s *nodes* (attachment points), and it is considered to be robust if the cable shape is roughly the same regardless of previous traveling path, or approach direction. To be *roughly* the same means in practice to achieve cable shapes that may differ up to a certain maximum distance from each other, regardless of approach direction.

To find a method for determining robustness of cable configurations, the relationship between the physical design of the robot and the cable configurations was examined. By knowing how the robot joints affect the cable configurations, an algorithm for mapping robustness could be developed. To verify the algorithm, a script was written in the programming language *Lua*. In IPS, Lua-scripts can be used for importing pre-defined cables, changing transformations of the cable’s nodes and saving information about the cable shape. To manipulate the node positions and orientations, the method of multiplying transformation matrices from Section 2.3 was used.

3.2.3 Cable wear cost function

The cable wearing that could not be avoided with the computationally efficient joint restrictions needed to be included and considered in the path planning algorithm. Since the already existing path optimization in IPS was subject to several objectives, like e.g. time minimization, smoothness and energy consumption, the introduction of the cable wear minimization needed to be done without removing these other objectives. Therefore the most suitable implementation was to construct a cost function, giving increasing cost with increasing cable wear. For this implementation the wearing causes needed to be translated to and expressed in measurable quantities.

Because of the high complexity of the optimization problem, being both highly non-linear and non-convex, the existing algorithm does not search for a *globally* optimal solution [12]. Instead the algorithm starts by finding a *nominal* path, to then be *locally* optimized. It is in this local optimization that the cable wear was to be considered, together with the other criteria (time, smoothness etc). Since there are typically many different solutions to the path planning problem, of which many have similar cycle time, the algorithm might find a path with less cable wear but without a radical increase in cycle time.

3.3 Test case evaluation

In order to evaluate the derived methods, a stud welding robot was chosen as a test case. The chosen robot was from the station that can be seen in the front page picture of this thesis. The stud welding robot is problematic when it comes to cable wear, since the dresspack has no retracting/feeding unit to pull back or feed out the cable when the robot moves. Instead the cable hangs down from the robot, increasing possibility of collision and twisting around the robot arm. To verify the functionality of the derived methods they were evaluated individually.

To get the cable to resemble the reality as much as possible, the cable material parameters were acquired through data acquisition from a similar *real* robot and cable setup at a stud welding station at Volvo Cars. By measuring mass and length of the cable, the length density could be calculated as $\rho_l = m/l$. By then moving the robot to four different poses and taking pictures, the pictures could be compared to the simulation and the remaining parameters tuned to get the model to resemble the reality as good as possible. The dimensions and material parameters that were chosen are presented in Table 3.1.

Parameter	Value
Length	1740 mm
Radius	30 mm
Length density	1.25 kg/m
Bending stiffness	0.1 Nm^2
Tensile stiffness	1400 N
Torsional stiffness	0.1 Nm^2

Table 3.1: Cable dimensions and material parameters.

3.4 Summary

The methodology of this thesis consists of three major parts: finding causes for cable wear, methods for implementing solutions and finally a test case evaluation. With these three steps the cable wear will both be identified and dealt with through three different methods. These methods handle the topics of: joint restrictions, robust cable configurations and cable wear cost function, where the first and the last method handles the actual cable wear reduction, and the second deals with the issue of finding non-robust configurations. The three methods together with the test case evaluation represent the outcome of this thesis, and they are all derived, evaluated and presented in **Chapter 4, Proposed Solution and Results**.

Chapter 4

Proposed Solution and Results

This chapter presents a proposed solution as well as acquired test results of this Master's Thesis. First, a presentation of the three sub-results is given. The sub-results include the topics of **joint restrictions**, **robust cable configurations** and **cable wear cost function**. After this the three sub-results are tested on a selected test case, upon which results are presented in **test case evaluation**.

4.1 Joint restrictions

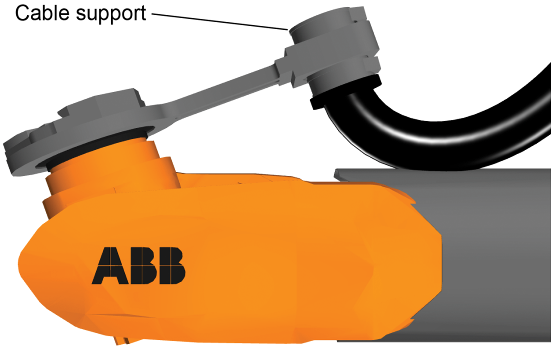
A computationally efficient way to avoid bad cable behavior is to use joint restrictions. By restricting the values of the robot joints in a clever way, some bad movements and poses for the cable can be completely accounted for. The reason why this method is computationally efficient, in relation to path planning robot and cable, is that a simple condition check is done, upon which a joint value is *allowed* if the condition is true and *prohibited* if it is false.

One example of bad behavior is if the cable twists too much around the robot arm, which leads to high stress and possible snapping of the cable. To prevent this the robot simulators at Volvo Cars use a "rule of thumb" suggesting that the absolute value of the angle sum of joints four and six must not exceed 270°, i.e.

$$| j_4 + j_6 | \leq 270^\circ \tag{4.1}$$

Inequality (4.1) is a simple condition check, where a given pair (j_4, j_6) is allowed *if and only if* (4.1) evaluates to boolean *true*.

Another bad behavior is when joints five and six are arranged in such a combination, that the joint six cable support comes too close to the robot arm (see Figure 4.1(a)). This results in the cable being crushed against the arm. To avoid this it is meaningful to have a certain clearance from the tip of the cable support down to the robot arm. This can be done by restricting the allowed space for joint five.



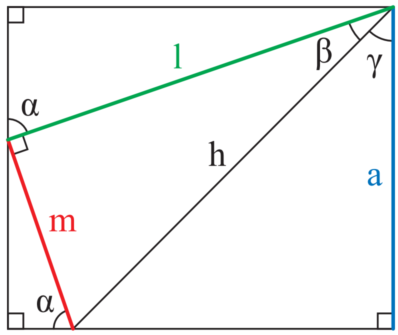


Figure 4.1: (a) j_5 at critical angle, crushing the cable against the arm. (b) Corresponding geometric figure.

The restriction on joint five is dependent on the angular value of joint six. For some angles of joint six, joint five can be changed arbitrarily without endangering any crushing of the cable, whereas for others a restriction is needed. One way to think of it is that it is only at certain *zones* of joint six that joint five must be restricted. Assuming joint six is in one of these zones, the restriction on joint five will depend on a function of joint six. This is due to the geometry of the robot arm (approximated as a cylinder). Also, since the cable support can be physically mounted in different initial angles relative to joint six, the zones will depend on this mounting angle. Figure 4.2 gives an example of the forbidden zones for joint five, as a function of joint six.

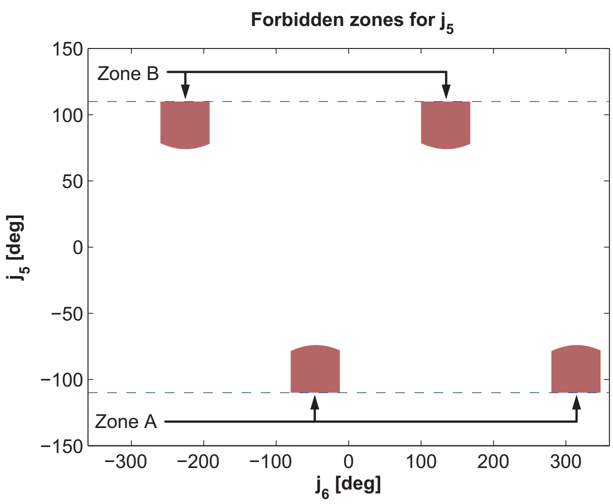


Figure 4.2: Forbidden zones for joint five, with the cable support mounted in a −45° angle (measured CW from joint six zero angle).

To determine the restriction function for joint five, the physical design of the robot is examined. Figure 4.1(b) shows a geometrical sketch of the robot arm in Figure 4.1(a), with m being the length of link five, l the length of the cable support and a the distance from the tip of the support down to the centerline of link four. The angle α is directly related to the value of joint five; $\alpha = -j_5$ when j_6 is in zone A and $\alpha = j_5$ when j_6 in zone B. The distance a can be approximated as

$$\begin{aligned} a &\approx r\,\cos(j_6 + p) + c && \text{if } j_6 \text{ in zone A} \\ a &\approx r\,\cos(j_6 + p + \pi) + c && \text{if } j_6 \text{ in zone B} \end{aligned}$$

where the parameters are described in Table 4.1. With explicit expressions for l, m and a, α can now be written as

$$\begin{aligned} \alpha &= \beta + \gamma \\ &= \cos^{-1}\left(\frac{l}{h}\right) + \cos^{-1}\left(\frac{a}{h}\right) \\ &= \cos^{-1}\left(\frac{l}{\sqrt{m^2 + l^2}}\right) + \cos^{-1}\left(\frac{a}{\sqrt{m^2 + l^2}}\right) \end{aligned} \tag{4.2}$$

With a in (4.2) being a function of j_6, and with the relation $j_5 = -\alpha$ (zone A) and $j_5 = \alpha$ (zone B), joint five can now be restricted with a function of joint six. Zone A will yield a *lower bound* on j_5 and zone B an *upper bound*, due to the physical design of the robot. The restriction on joint five becomes

$$j_5 \geq f_A(j_6) = -\cos^{-1}\left(\frac{l}{\sqrt{m^2 + l^2}}\right) - \cos^{-1}\left(\frac{r\,\cos(j_6 - p) + c}{\sqrt{m^2 + l^2}}\right) \quad \text{if } j_6 \text{ in zone A} \tag{4.3}$$

$$j_5 \leq f_B(j_6) = \cos^{-1}\left(\frac{l}{\sqrt{m^2 + l^2}}\right) + \cos^{-1}\left(\frac{r\,\cos(j_6 - p + \pi) + c}{\sqrt{m^2 + l^2}}\right) \quad \text{if } j_6 \text{ in zone B} \tag{4.4}$$

where the parameters are summarized in Table 4.1.

Parameter	Description
r	Radius of link four (approximated as a cylinder).
m	Length of link five.
l	Length of cable support.
c	Required clearance to link four.
p	Angle for cable support. Measured CW from j_6 zero angle $[-180°, 180°]$.

Table 4.1: Parameters for joint restriction function.

Since the robot arm is approximated as a cylinder, the restriction functions in (4.3) and (4.4) work best for (roughly) cylindrical robot arms. Although the upper arm of the robot in Figure 4.1(a) is roughly cylindrical, the wrist close to link five is not. This is why the cylindrical approximation can be replaced by a cuboid approximation. With the cuboid approximation, the restrictions on joint five become

$$j_5 \geq -\cos^{-1}\left(\frac{l}{\sqrt{m^2+l^2}}\right) - \cos^{-1}\left(\frac{r+c}{\sqrt{m^2+l^2}}\right) = -K \qquad \text{if } j_6 \text{ in zone A} \qquad (4.5)$$

$$j_5 \leq \cos^{-1}\left(\frac{l}{\sqrt{m^2+l^2}}\right) + \cos^{-1}\left(\frac{r+c}{\sqrt{m^2+l^2}}\right) = K \qquad \text{if } j_6 \text{ in zone B} \qquad (4.6)$$

where K is a (pre-calculated) constant for a given clearance c and r is now half the height of the cuboid. The cuboid approximation is more accurate than the cylindrical for shorter cable supports and/or large clearance, and also requires even less calculation processing. To know which approximation to use, one can examine the tip of the cable support when j_5 is at its critical angle. If the tip is above the cylindrical part of the robot arm, the cylindrical approximation is to be used, and if above the wrist the cuboid approximation is chosen.

An easily implemented method for including all the joint restrictions is to combine them into a single logical expression. In this section only two different types of restrictions have been dealt with, but the implementation also works for a larger number of restrictions. Using the cylindrical approximation for the restriction on joint five, the joint restrictions from (4.1), (4.3) and (4.4) can be combined into the logical expression

$$[\mid j_4 + j_6 \mid \leq 270°] \wedge [[j_6 \text{ in zone A}] \wedge [j_5 \geq f_A(j_6)] \vee [j_6 \text{ in zone B}] \wedge [j_5 \leq f_B(j_6)]] \qquad (4.7)$$

which can easily be implemented in computer code. The expression must evaluate to boolean *true* for any given set of joint angles (j_4, j_5, j_6), i.e. a given joint configuration is allowed *if and only if* (4.7) is *true*.

4.2 Robust cable configurations

When determining robustness of cable configurations, the most straightforward way might be to attach a dresspack to a robot and then test robustness for all possible combinations of the joint angles $(j_1, j_2, j_3, j_4, j_5, j_6)$. However this would result in too many configurations to test (about 1.13×10^{15} for an angular resolution of 1°). In this section it is shown that only a subset of these configurations need to be tested, and an algorithm for testing the configurations is derived. The developed method is also verified by finding non-robust configurations for two different test cables.

To find out which of the configurations that can be excluded from the testing, one must first examine how the physical design of the robot affects the movement of the cable.

An industrial robot is limited to movement in six degrees of freedom. Considering only the end part of a cable, that goes from link three to link six, the movements of joints one, two and three will transform all cable nodes in a uniform fashion, i.e. as one rigid unit. As an example, consider the robot and cable in Figure 4.3. Here the two rightmost nodes are rigidly connected to link three, and the leftmost node to link six. In this case joints one, two and three will move all cable nodes as one unit, whereas joints four, five and six will only affect the leftmost node.

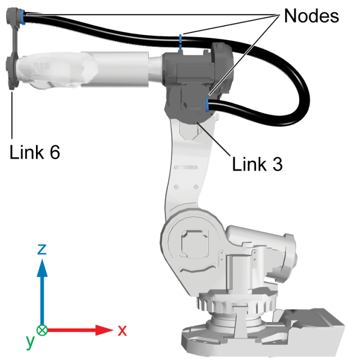


Figure 4.3: Robot with cable attached to links three and six.

Intuitively, a pure translation of the cable in Cartesian *(x,y,z)* will not affect the cable shape. Similarly, rotating about the *z-axis* in Figure 4.3 will also not affect the cable shape. Assuming the rotation in z is locked to the position in Figure 4.3, the cable will only be able to rotate about the *y-axis* (since the robot cannot tilt sideways about the *x-axis*). With these limitations, it is only of interest to study configurations as a result from

- rotation of the entire cable about the *y-axis* (R_y).
- movement of leftmost node by manipulating joints four, five and six (j_4,j_5,j_6).

Since the only parameters determining the positions of the cable nodes are the ones just described, a cable configuration can be defined by the configuration vector

$$\mathbf{v} = (R_y,j_4,j_5,j_6) \tag{4.8}$$

The condition is that the cable and robot setup is as above, i.e. that joints one, two and three move all cable nodes as one unit, and that the robot is either floor- or ceiling mounted.

With the desired configurations identified, an algorithm for measuring robustness for a configuration can now be developed. As described in Section 3.2.2, a robust configuration has roughly the same cable shape regardless of previous traveling path. Therefore, to determine robustness the cable needs to be moved to a configuration along different paths. To keep down computational effort, a total of two paths are selected for the robustness testing. This resulted in Algorithm 1. The idea is to first move the cable nodes to the configuration to be tested, and then save the cable shape by saving N_s (x,y,z)-sample points along the cable segment. By then *decreasing* the values of the configuration vector $\mathbf{v}$, followed by increasing back to the current configuration, a *negative* movement has been made. Ideally the cable shape should now be exactly the same as before the movement, and any too large deviations implies non-robustness. By then doing a *positive* movement in a similar manner, two cable shapes have been acquired to be compared to the initial shape. If, for any sample point, the distance to corresponding sample point in the initial shape is greater than distance threshold d_{thresh}, the configuration is non-robust.

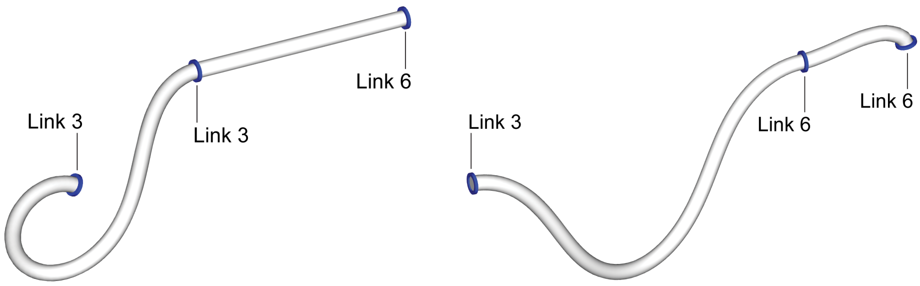
Algorithm 1 Configuration robustness mapping

$\mathbf{V} \leftarrow \emptyset$ ▷ Init non-robust configurations
for $i \leftarrow 1, N$ **do** ▷ For all N configurations
    $moveNodes(\mathbf{v}(i))$ ▷ Move all nodes to config i
    $S_0 = getShapeData()$ ▷ Store initial shape
    $moveNodes(\mathbf{v}(i) - (20°,100°,100°,100°))$ ▷ Do *negative* movement
    $moveNodes(\mathbf{v}(i))$ ▷ Move back to config i
    $S_1 = getShapeData()$ ▷ Store shape
    $moveNodes(\mathbf{v}(i) + (20°,100°,100°,100°))$ ▷ Do *positive* movement
    $moveNodes(\mathbf{v}(i))$ ▷ Move back to config i
    $S_2 = getShapeData()$ ▷ Store shape
    for $j \leftarrow 1, N_s$ **do** ▷ For all cable segment sample points
        if $d(S_0(j), S_1(j)) \geq d_{thresh}$ **or** $d(S_0(j), S_2(j)) \geq d_{thresh}$ **then**
            $\mathbf{V} \leftarrow \mathbf{V} \cup \mathbf{v}(i)$ ▷ Save non-robust config
        end if
    end for
end for

To verify that the algorithm produced a reasonable mapping of non-robust configurations, it was tested with two different cables. To resemble a real case scenario, one spot welding cable and one stud welding cable were used (cable A and B, respectively). Figure 4.4 shows how the two cables were attached to the robot links. The configurations were achieved by taking all possible permutations of

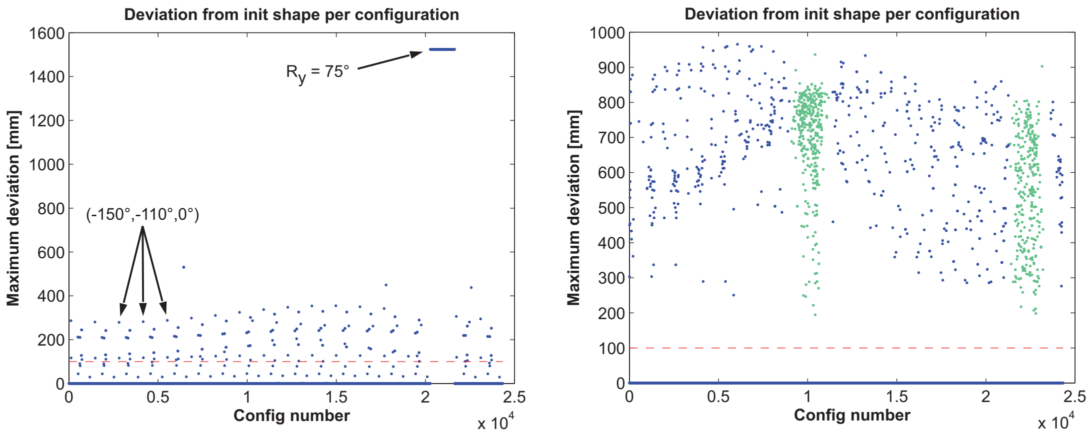
- R_y from $-225°$ to $115°$ increment $20°$
- j_4 from $-180°$ to $180°$ increment $30°$
- j_5 from $-110°$ to $110°$ increment $30°$
- j_6 from $-180°$ to $180°$ increment $30°$

This yielded 24,336 unique configurations. For each configuration the maximum deviation from the initial shape was calculated using Algorithm 1 and then plotted (see Figure 4.5). With the deviation threshold $d_{thresh} = 100$ mm, a total of 6% (A) and 5% (B) of all configurations were found to be non-robust. For cable A some patterns for non-robust configurations could be seen, e.g. the combination $(j_4, j_5, j_6) = (-150°, -110°, 0°)$, which always yielded non-robust configurations (except for $R_y = 55°$). Also, $R_y = 75°$ yielded non-robust configurations, regardless of the other joint values. However for cable B no definite pattern could be established, since the occurrence of non-robust configurations was more random than for cable A. Although no definite pattern was found, an increase in the amount of non-robust configurations for R_y close to $\pm 90°$ could be seen. For these configurations 17% could be considered to be non-robust.



(a) (b)

Figure 4.4: (a) Cable A and (b) cable B, with marked link attachments.



(a) (b)

Figure 4.5: Simulation results for (a) cable A and (b) cable B. The red dashed line indicates the threshold between robust and non-robust configurations. The green point clusters in (b) indicate $R_y = \pm 90°$.

4.3 Cable wear cost function

Cables wearing out involves several different types of wearing factors, and as described in Section 3.1 this thesis focuses on the ones that are affectable by the robot path. Interviews with simulation experts and on-line robot programmers revealed that the main factor is contact with static geometry (e.g. the car body). When hitting a sharp metal edge at high speed this type of damage is amplified, but since this thesis is delimited to only include static cable simulations, this phenomenon will not be accounted for. However, this effect can be reduced by increasing the clearance threshold when doing the automatic path planning.

Another type of wearing that was shown from the interviews and from [1] is bending of the cable. Excessive bending can lead to insulation cracking and damage to the cable [15], and one major cause for this is bad robot paths. By avoiding robot paths with excessive bending this kind of damage can be reduced. A special siuation identified in [1] is improper mounting of the cable that may be damaging independent of the path. This could easily be detected by static analysis, and does not require path optimization.

A third problem that was found during one of the study visits at Volvo Cars is when the cable gets stuck somewhere on the robot arm, and then gets tugged by the robot. This introduces high stress on the cable and could lead to severe damage at the attachment points or on the cable itself. This kind of problem might arise when the cable hangs loose, without any retracting/feeding unit to pull back or feed out cable when the robot moves.

For the three major wearing causes; contact, bending and tugging, to be minimized, they need to be detectable and measurable in IPS. As for the contact, the shortest distance to any static geometry can be used as a measure. This being zero is equivalent to being in contact, and anything else can be directly related to a desired clearance. Since there will always be some amount of uncertainty in the simulation compared to the reality, it is desirable to have a certain clearance between the cable and surrounding geometry. Also, as previously mentioned, clearance is a way to take uncertainties due to dynamical effects into account.

The deformed shape from bending a cable can be determined completely by the bending radius R [16]. A very tight bend is undesirable and can be detected by a very small bending radius. To avoid tight bending but allow smoother bending, the curvature is a natural bending measure, being the inverse of the radius ($\kappa = 1/R$). For a bending radius going to zero the curvature will go to infinity, which used as a cost will prevent very tight bends.

To measure or detect tugging, the tension force of the cable can be used. The tension force is the longitudinal force in the cable and in normal cases, i.e. when the cable suffers no drastical pulling, it is rather small. However, when the cable gets stuck and tugged the tension force grows fast with increased tugging.

With acquired measures of cable wear, it is possible to construct a cost function giving increasing cost with increased wearing. Since the already existing path optimization in IPS utilizes minimization of other objectives like e.g. time, it is suitable to include the cable wear cost function as a term among these other costs. The overall goal of the optimization is to optimize the path by tuning a set of N abstract configuration vectors defining the path. These configuration vectors are considered to be the variables of the optimization, and together they form the path

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_\mathbf{1} \\ \gamma_\mathbf{2} \\ \vdots \\ \gamma_\mathbf{N} \end{bmatrix} \tag{4.9}$$

where $\gamma_\mathbf{i}$ is configuration vector i. Note that $\gamma_\mathbf{i}$ is not the same configuration vector as $\mathbf{v}$ in Section 4.2, but rather a more general vector with all joint values included.

When constructing the cost function it is not suitable to use the acquired measures directly, because of reasons as follows. Since the cable is typically not straight under normal bending circumstances, the curvature must have a *lower threshold* for the bending penalty to take effect. Similarly for the shortest distance measure, being further away than a specified clearance must not result in any penalty. This is why the shortest distance measure needs an *upper threshold*, which is equal to the desired clearance.

To ensure that large violations of the cable wear factors result in extra penalization, all factors are squared. By then multiplying each factor with a cost weight, the optimization can be tuned to take each factor into more or less consideration. Summing the three weighted terms together, the cable wear cost function for configuration $\gamma_\mathbf{i}$ becomes

$$C(\gamma_\mathbf{i}) = \underbrace{w_\kappa \cdot \max(0,\kappa(\gamma_\mathbf{i}) - \kappa_0)^2}_{C_\kappa(\gamma_\mathbf{i})} + \underbrace{w_F \cdot F(\gamma_\mathbf{i})^2}_{C_F(\gamma_\mathbf{i})} + \underbrace{w_d \cdot \min(0,d(\gamma_\mathbf{i}) - c)^2}_{C_d(\gamma_\mathbf{i})} \tag{4.10}$$

where the parameters are presented in Table 4.2 and

- $\kappa(\gamma_\mathbf{i})$ is the largest curvature of the cable,
- $F(\gamma_\mathbf{i})$ is the largest tension force,
- $d(\gamma_\mathbf{i})$ is the shortest distance to any static geometry.

Parameter	Description
κ_0	Smallest curvature for the curvature penalty to take effect.
w_κ	Cost weight, curvature.
w_F	Cost weight, tension force.
w_d	Cost weight, shortest distance.
c	*Desired* clearance to surrounding geometries.

Table 4.2: Parameters for cable wear cost function.

For better understanding of how the three terms $C_\kappa(\gamma_\mathbf{i})$, $C_F(\gamma_\mathbf{i})$ and $C_d(\gamma_\mathbf{i})$ contribute to the overall cost, they have been plotted individually as functions of their respective wearing factors (see Figure 4.6). Curvatures that are $\leq \kappa_0$ are neglected, and curvatures going to infinity result in the cost going rapidly to infinity. Tension forces going to infinity gives cost going rapidly to infinity, and the distance cost is always in the interval $[0,\ w_d \cdot c^2]$. By not having a distance cost going to infinity for distance going to zero, contact with surrounding geometry is *allowed*, however still penalized.

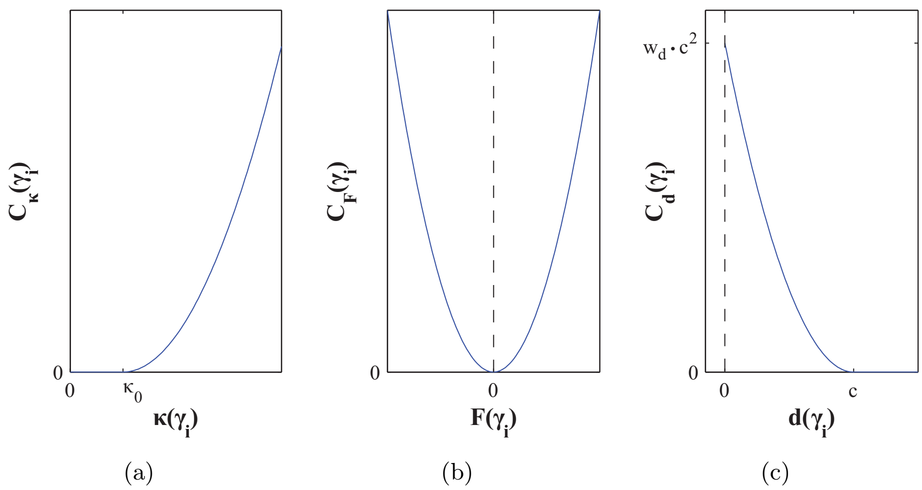


Figure 4.6: The three cost terms in the cable wear cost function (a) $C_\kappa(\gamma_\mathbf{i})$, (b) $C_F(\gamma_\mathbf{i})$ and (c) $C_d(\gamma_\mathbf{i})$.

4.4 Test case evaluation

To verify the functionality of the methods, each method was evaluated individually on a test case. As described in Section 3.3, the chosen test case is a stud welding robot from the station on the thesis front picture.

To acquire joint restrictions, the physical design of the robot was examined. In this case the joint restriction for the cable support is not necessary, since j_5 and j_6 can be changed arbitrarily without endangering any crushing of the cable. However the restriction from Inequality (4.1) can be used with some modification to account for a different initial mounting position of the dresspack. Since joints four and six are both turned 180° for the home configuration, the joint restriction becomes

$$j_4 + j_6 \geq 90^\circ \tag{4.11}$$

Moving on to the cable configurations, a mapping was done using Algorithm 1 to acquire the non-robust configurations. The joint angles were achieved by taking all possible permutations of

- R_y in 20 steps evenly distributed around 0° from −180° to 180°
- j_4 in 15 steps evenly distributed around 0° from −180° to 180°
- j_5 in 15 steps evenly distributed around 0° from −110° to 110°
- j_6 in 15 steps evenly distributed around 0° from −180° to 180°

This resulted in 67,500 different configurations. With a non-robustness threshold $d_{thresh} =$ 100 mm, a total of 5,743, or 9%, can be considered as non-robust. Figure 4.7 shows how the amount of non-robust configurations varies with R_y. As in the case with cable B in Section 4.2, the amount increases for R_y close to ±90°.

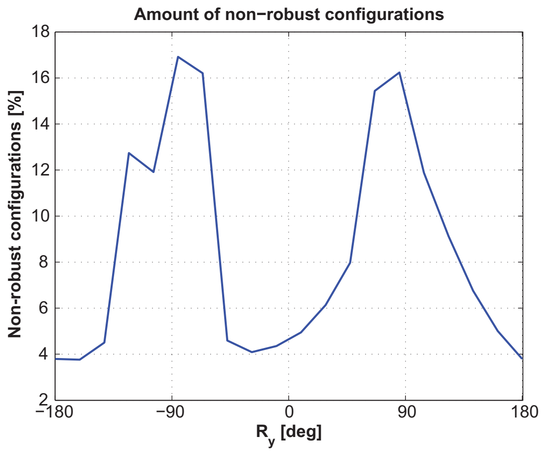


Figure 4.7: Amount of non-robust configurations for R_y.

To evaluate the cable wear cost function, three different scenarios were constructed for curvature, tension force and distance costs. For each cost term evaluation, the cost weights for the two remaining terms were put to zero. Each scenario resulted in a plot, comparing performance between the two cases; *with* and *without* cable wear costs. These plots are presented in Appendix A.

For the curvature evaluation, a scenario where one end of the cable was attached to the robot tool and the other statically fixed in mid-air, was constructed (see Figure 4.8). The robot was to move between the two configurations *Home* and *Weld*, at which both yielded rather small curvature for the cable. However in the case without curvature cost, the cable underwent sharp bending in the middle of the path when the tool passed close to the static cable node. By introducing the curvature cost from Equation (4.10), with $w_\kappa = 1$ and $\kappa_0 = 8$ [1/m], the path was altered to allow smooth bending but try to avoid bending where $\kappa \geq 8$ [1/m]. This resulted in a 62% decrease in maximum curvature during the path.

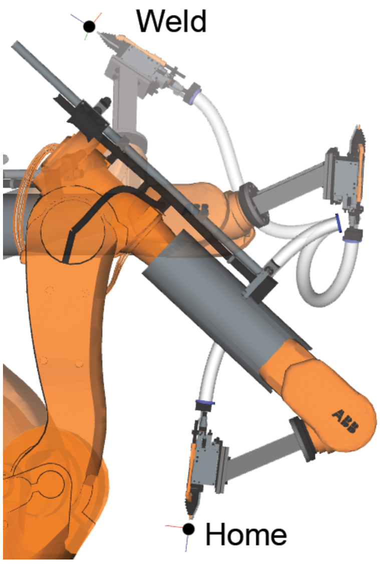
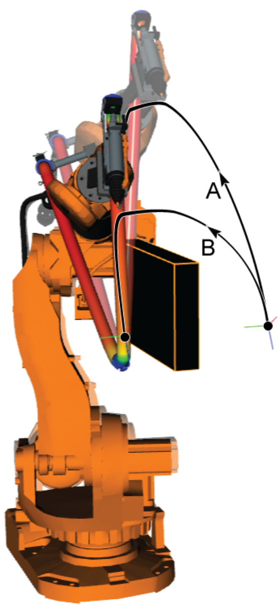


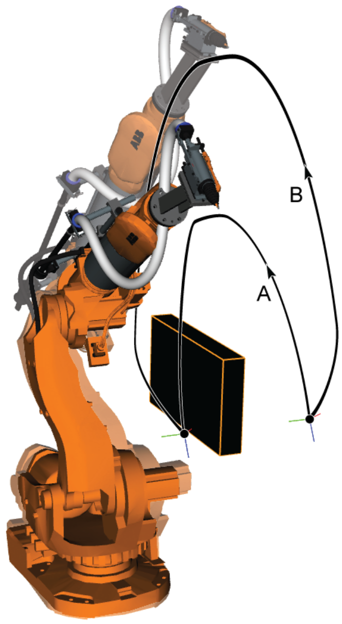
Figure 4.8: Curvature test case. The robot moves from *Home*, via *Weld* and then back to *Home*.

In the second scenario the tension force is evaluated. In this scenario the robot was to move between two points without colliding with a cuboid obstacle (see Figure 4.9(a)). By having a cable node firmly attached underneath the cuboid, movement of the robot above the cuboid introduces high tension force. With a desired robot clearance of 500 mm the path planning algorithm finds a path high above the cuboid, when no force cost is considered (path (A) in Figure 4.9(a)). However when the force cost is introduced with $w_f = 1 \times 10^{-5}$, the path gets closer to the cuboid ((B) in Figure 4.9(a)). This is because it is less costly to violate the clearance than to have a high tension force. With the force cost the maximum tension force decreased by 61%.

The third scenario is for the distance cost evaluation. Again the robot was to move between two points, both with and without the distance cost (see Figure 4.9(b)). Without the cost, the robot moves closer to the obstacle (path (A)) and with the cost, the robot moves both higher above and further away from the sides of the obstacle (path (B)). The cost parameters for path (B) were $w_d = 100$ and $c = 1$ [m].



(a)



(b)

Figure 4.9: (a) Tension force test case. Path (A) is without cost and path (B) is with cost. The cable is subject to a tension force analysis with hot and cold colors for higher and lower tension force, respectively. (b) Shortest distance test case. Path (A) is without cost and path (B) is with cost.

4.5 Summary

This chapter has presented the results and outcome of this thesis, consisting of two methods for cable wear reduction, one method for dealing with non-robust configurations and finally a test case evaluation. The two methods for reducing cable wear involves one computationally efficient method, *joint restrictions*, and one computationally more cumbersome method, *cable wear cost function*. The mapping of non-robust configurations is a difficult but essential part of the overall solution, and the test case evaluation shows how the methods perform individually. Now a discussion about the results as well as conclusions and suggestions for future work will be given in **Chapter 5, Discussion and Conclusion**.

Chapter 5

Discussion and Conclusion

In this chapter a discussion about the result is given. Also, conclusions and suggestions for future work are presented. The structure is built on the three sub-results from the result chapter; **joint restrictions**, **robust cable configurations** and **cable wear cost function**, and then ended with some summarizing **concluding remarks**.

5.1 Joint restrictions

The main reason for restricting the joint values to avoid cable wear is the computational efficiency. By avoiding any collision checking and cable simulation, and simply just checking the joint values, this is a highly efficient method. It is also beneficial considering implementation, since it is easy to check whether a certain joint value fulfills a criterion, and to restrict it if it doesn't. There is however a (small) drawback with this method, and that is that it is not very general, but valid only for specific robot applications. For example, the restriction functions for joint five (4.3), (4.4), (4.5) and (4.6) only work for robots with the specified cable support mounted on the tool plate. For a robot application with different dresspack mounting, new options for joint restrictions need to be examined.

Since the computational gains are so great for this method, it is worth spending some extra time trying to find joint restrictions. When assessing a new dresspack and robot setup, the first thing to check should be if there are any joint restrictions that can be implemented to minimize wearing. With restrictions dependent on mounting parameters, like e.g. the cable support angle, cable wear can be minimized with high computational efficiency and with some generality and flexibility.

5.2 Robust cable configurations

The justification for developing the robustness mapping algorithm was the fact that non-robust cable configurations is a big problem. If one of the tasks, i.e. one of the robot configurations for a certain weld point, results in a non-robust cable configuration, the sequencing step in the working procedure will not be feasible. This means that it will not be possible to change the order of the tasks or to alter the paths leading to the non-robust configuration. Ignoring the non-robust configuration will make any cable wear consideration in the path optimization inapplicable. If a non-robust configuration is found among the robot tasks, one easy fix to this problem is to determine the task order and perform the path planning for all tasks prior to the configuration, *without* cable wear consideration. This will give usable but somewhat inferior results.

Since no previous research was found on cable robustness mapping, the procedure for developing the algorithm was based on knowledge of the problem alone. Although verified with two different test cases, the algorithm can not be considered to be completely validated. This is because some issues still need to be dealt with, like e.g.

- the definition for a non-robust configuration might be too vague.
- the choice of distance threshold d_{thresh} is not trivial.
- it has not been shown that the two paths in the algorithm are enough for determining non-robustness.

This is why one suggestion for future work is to deal with these issues and refine the algorithm. Since the definition for non-robustness implies having to test *all* possible traveling paths to a configuration, a better definition is probably the first thing to investigate. By proving some kind of completeness of the definition, e.g. having a high probability of rightly classifying robustness by moving along a small number of "bad enough" paths, the robustness mapping should be implementable in a commercial context.

Having pointed out the flaws of the robustness mapping, this does not reject the acquired results from the work in this thesis. Although the developed algorithm could not determine if a configuration is *robust*, it could determine *non-robustness* for certain configurations. With a large enough distance threshold, the algorithm could conclude that some configurations definitely were non-robust. One example is the robustness mapping for cable A, where $R_y = 75°$ yielded shape deviations of about 1500 mm. For these configurations non-robustness can be concluded without any doubt.

Another advantage of the algorithm is that the choice of the distance threshold, although maybe not trivial, can be directly related to a clearance measure. By allowing the choice to the user of the algorithm, he or she can set the threshold to a desired clearance to surrounding geometry, allowing for larger or smaller uncertainties in the simulation.

5.3 Cable wear cost function

For the cable wear cost function to be a valid indicator of cable wear, it is essential that the acquired wearing factors make a good representation of the reality. The determination of the wearing factors was done both through interviews with matter experts and by examining the field studies from Volvo Cars in [1] and [2]. This proceeding gave precise and practical knowledge from highly experienced sources, and with the experts from Volvo Cars having many years of experience from a plant with as much as 850 robots [17], the wearing factors should be applicable for other industries as well.

The evaluation of the cost function does highlight some interesting behaviors. For example, when the curvature was minimized the path was not altered for curvatures $< \kappa_0$. At first glance it might therefore seem strange that the curvatures *with* and *without* cost in Figure A.1 do not follow the same profile up to the threshold κ_0. There is however an explanation for this. When the path is optimized, the cost function is evaluated at configurations distributed around a nominal path. If the initial distribution of configurations results in a non-zero cost at the beginning of the path, that configuration will be altered to give a lower cost. Also, having altered one point might lead to a chain effect where other points along the path are altered, to optimize other objectives (like e.g. smoothness of the path). This also holds for the clearance threshold in the distance cost term.

Another interesting behavior can be seen in the the tension force minimization. Here the optimization always resulted in robot paths that went *above* the obstacle. Although the tension force was in fact minimized, it would have been even less costly to take the longer path *underneath* the obstacle. For this path the tension force would never exceed 40 N, whereas for the acquired path it peaked at 450 N. The reason why the planner finds the path above the obstacle lies in the minimization method that is being used. First the planner finds the shorter path above the obstacle, and then locally alters the path to minimize the tension force. This is why the path will be moved locally and in small steps closer to the top of the obstacle, and the path underneath will not be found. Therefore, to deal with scenarios like this, it might be preferable to consider the cable wear minimization as more of a global problem. However, doing this might result in dramatically increasing computational effort, since the cable would have to be simulated more often in the optimization.

In the third scenario when the distance cost was considered, the optimization did result in larger clearance between the cable and obstacle, both to and from the ”weld point”. The distance cost is a good way to account for any uncertainties in the modeling of the cable, since big uncertainties can be compensated for by having larger cable clearance. Worth to mention is also that the distance cost might not always result in non-violated clearance, even for large cost weights. One explanation for this is that the evaluation of the cost function is done at quite random sample points along the path. This leads to fast jumps between configurations possibly far away from each other, resulting in cable shapes that are hard to determine. The jumping between configurations might lead to different paths from point A to B, than from B to A. To solve this the sample points must be evaluated in some kind of order, that gives shorter and smoother jumps for the cable.

To have the cable wear cost function fully working in a commercial context, some further testing is needed. For future work it is suggested to investigate improvements in the cost terms for a better cable wear estimate. One example of improvement is normalization of the terms, so that the cost weights are equally significant. In other words, $w_\kappa = w_F = w_d$ should give roughly equal penalization of all three terms.

5.4 Concluding remarks

Excepting the limitations discussed in this chapter, the evaluation of the test case verifies the individual functionality of the proposed methods. Thus the first question from Section 1.2 is answered *(Do the methods perform as intended?)*. With the implementation of the discussed improvements, the methods should work to minimize dresspack wear for a path from one point to another, and then to a third. With this optimization working, the paths for an entire sequence of points can be optimized for cable wear, and with this a multi-robot station or even an entire robot line could be optimized. To summarize and answer the second question from Section 1.2 *(In excess of these methods, what more is needed for a commercially acceptable solution?)*, the following is needed for a commercially acceptable solution:

- Find a better definition for a non-robust configuration and re-verify functionality of the robustness mapping.
- Implement an evaluation order for the configurations that gives smoother jumps for the cable.
- Investigate and implement improvements in the cable wear cost terms, like e.g. the normalization of the cost weights.
- Validate the combined effort of the methods.

Having verified the functionality of the methods after the improvements, a multi-robot test case can be constructed for a complete validation. After validating that the dresspack wear is in fact reduced, a fully implementable solution will have been derived.

Bibliography

[1] Eriksson, U. *Poor Life Length of Cable Packages*. Volvo Cars, Göteborg. 2005.

[2] Eriksson, U. *Excessive Exchange of Hose Packages*. Volvo Cars, Göteborg. 2005.

[3] REIKU. *Robotic cable management systems*. http://www.reikuna.com/catalogue-library (accessed May 3rd 2013). File: REIKU Robotic Cable Management.pdf.

[4] Igus. *The Less-is-More Approach to Robotic Cable Management*. http://www.igus.com/wpck/default.aspx?Pagename=The_Less_is_More_Approach_to_Robotic_Cable_Management&C=US&L=en (accessed May 3rd 2013).

[5] Drives & Controls. *Wireless sensors replace vulnerable robot wiring*. http://www.drives.co.uk/fullstory.asp?id=1112 (accessed May 3rd 2013).

[6] Saboori, B; Saboori, B; *Development of roller hemming process in automotive industry - Saab Automobile AB*. Chalmers University of Technology. Göteborg. 2009.

[7] EUnited Robotics. *IRB 6600ID*. http://www.eu-nited.net/robotics/index.php?idcat=81&idart=121 (accessed May 6th 2013).

[8] ABB. *DressPack Application Equipment & Accessories*. http://www.abb.com/product/seitp327/f10cb8df3c6d8fe4c125745000302ba2.aspx (accessed May 6th 2013). File: Data sheet DressPack (pdf).

[9] Linn, J.; Stephan, T.; Carlson, J.; Bohlin, R. ”Fast simulation of quasistatic rod deformations for VR applications” in *In Progress in Industrial Mathematics*. 2006. p. 247-253.

[10] Hermansson, T; Bohlin, R; Carlson, J; Söderberg, R. ”Automatic Assembly Path Planning for Wiring Harness Installations” in *Journal of Manufacturing Systems*. 2013.

[11] McKerrow, P.J. *Introduction to Robotics*. Addison-Wesley Publishing. Singapore. 1991.

[12] LaValle, S.M. *Planning Algorithms*. Cambridge University Press. Cambridge. 2006.

[13] Bohlin, R. *Motion Planning for Industrial Robots.* Ph.D. Thesis. Chalmers University of Technology and Göteborg University. Göteborg. 1999.

[14] Andréasson, N; Evgrafov, A; Patriksson, M. *An Introduction to Continous Optimization.* Studentlitteratur. Lund. 2005.

[15] John Whitfield. *The Electrician's Guide to the 17th Edition of the IEE Wiring Regulations BS 7671:2011 and Part P of the Building Regulations.* EPA. Saffron Walden. 2012.

[16] Krenk, S. *Mechanics and Analysis of Beams, Columns and Cables.* Springer. Berlin. 2001.

[17] Olsson, J. *Volvo bygger för nya hemmamarknaden.* `http://www.nyteknik.se/nyheter/automation/verkstadsautomation/article3545545.ece` (accessed May 6th 2013).

Appendix A

Cable wear test case plots

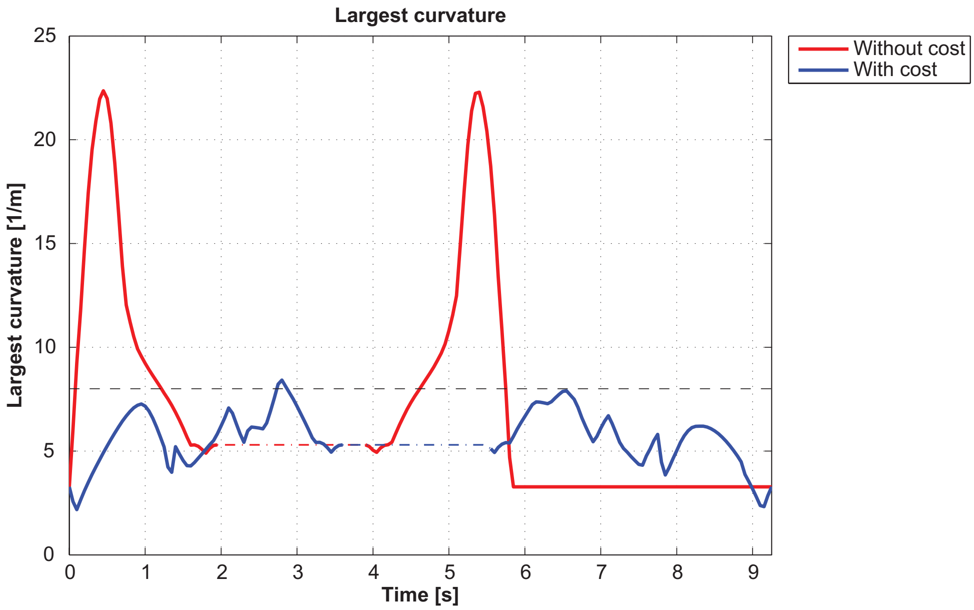


Figure A.1: Largest curvature for the cable, both with and without curvature cost. The dashed line indicates the curvature threshold $\kappa_0 = 8$ [1/m] and the dash-dotted sections indicate when the robot is welding. Note that the two cases are not comparable for a given time sample, since the path *with* cost takes longer time than the one *without*.

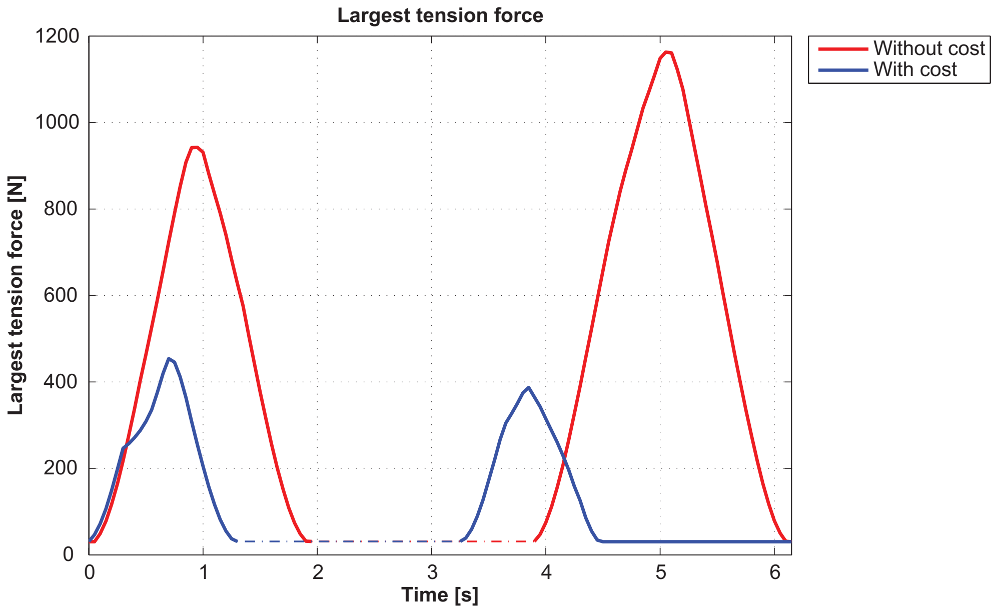


Figure A.2: Largest tension force for the cable, both with and without tension force cost. The dash-dotted sections indicate when the robot is welding. Note that the two cases are not comparable for a given time sample, since the path *without* cost takes longer time than the one *with*.

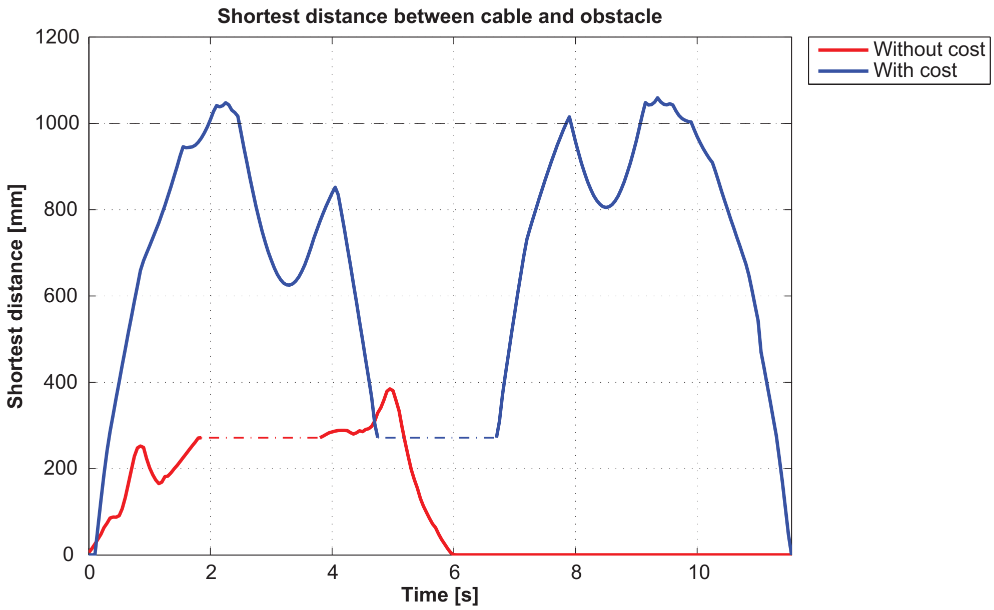


Figure A.3: Shortest distance between cable and obstacle, both with and without cable distance cost. The dashed line indicates the clearance threshold $c = 1$ [m] and the dash-dotted sections indicate when the robot is welding. Note that the two cases are not comparable for a given time sample, since the path *with* cost takes longer time than the one *without*.